Bypassing Captive Portals

and Limited Networks

Grant Bugher

http://perimetergrid.com

DEFCON 101 @ DEFCON 24

Who am I?

- Hacking and coding since the early 90's
- Working professionally in information security for the last 10 years
  - Developer, security tester, program manager, security engineer, security architect, consultant, educator – a bit of everything
  - Worked on IT, developer tools, programming languages & class libraries, online services, high-security datacenters, telecommunications/VoIP systems, application security consulting, SIEM deployment, retail systems
  - Currently a security engineer for a major cloud service
  - Also run Perimeter Grid, security blog & consulting service
- Prior speaker at BlackHat USA (2010) and DEF CON (22, 23) and a regular DEF CON attendee since DEF CON 16.

#perimetergrid

The research and opinions presented in this talk are my own.

They do not necessarily represent those of my employer.

Captive Portals and Limited Networks

- Primitive form of NAP (Network Access Protection)
  - Open network (e.g. Ethernet, DOCSIS, or open (unencrypted) WiFi)
  - Initial join allows access only to a limited web site (captive portal)
  - Limited website can authorize access to wider network (Internet)
- Commonplace
  - Every store/restaurant’s open WiFi
  - Hotel/airline Internet
  - Many corporate environments' guest networks
  - Some telecom networks (e.g. subscription hotspots)

Stupid Networking Tricks

- Not “real” NAP
  - No real authentication, just simple identifiers
  - No real encryption, just obfuscation
  - …no real security
- Enforcement at the gateway
  - Captive portal always accessible, as are some infrastructure services (DHCP, DNS, proxy config)
  - Either MAC filtering on the gateway or “authenticated” proxy
- Reliant on “obedient” network clients

Not Much Variety

- Chilispot
  - Open source captive portal gateway, built into OpenWRT & available on most Linxues
  - Requires web server for presenting captive portal
  - Requires RADIUS server if users are to be authenticated
- Everything's just Chilispot
  - Worldspot.net, HotspotSystem, Sputnik, HotspotExpress, Wifi-soft, Skyrove…
  - DD-WRT, OpenWRT, most commercial routers with hotspot capability
- Even if it isn't Chilispot… it still is
  - While the details vary, the enforcement mechanisms don't

Preparing Your Endpoint

Tunneling Traffic

- Tunneling is just moving one protocol via another
  - Usually encrypted (e.g. VPN and IPsec tunnels), but it doesn't have to be
  - Requires a server to act as the other "end" of the tunnel
- Need a protocol the captive portal won't block
  - HTTPS and SSH are sometimes unblocked on specific ports
  - DNS is almost always proxied out for us (DNS recursion)

Setting up a Server

- Need to have an Internet-accessible server to act as your tunnel endpoint
  - Any cheap VPS that gives full port control (not just web)
  - Cheap/free AWS EC2 or Azure Compute node
  - Your own home PC
- Multiple endpoints:
  - HTTPS proxy on 80, 443
  - SSH on 22, 3128 (squid default port)
  - Iodine on 53 with an NS record pointing at it somewhere
  - Be sure to open these ports on your EC2/Azure firewall if applicable

SSH Setup

- Any decent VPS will come with SSH enabled
- Edit /etc/ssh/sshd.config:
  - Add “Port 3128”… and any other ports you want. No limit on number.
  - Disable insecure logins while you’re at it

```
PasswordAuthentication no
RSAAuthentication yes
PubkeyAuthentication yes
```

- Ensure you have a public key in authorized_keys and on your portable machines

Iodine Setup

- On VPS:

```
sudo apt-get install iodine
sudo iodined -c -P password 172.16.0.0 subdomain -n publicip
```

- On DNS server:
  - Two custom records: one for the subdomain, one for the nameserver
  - Example:

```
ns.t.perimetergrid.com IN A publicip
t.perimetergrid.com IN NS ns.t.perimetergrid.com
```

  - Use short domain names if possible for efficiency – they go on every packet
  - Namecheap FreeDNS (free) or Amazon Route 53 (not free) works if you don't have a DNS server

HTTPS Proxy Setup

- Low value
  - Will not bypass most restricted networks and captive portals
  - Useful when on a network that allows web traffic out but not other traffic
- On VPS:
  - `sudo apt-get install squid3`
  - Replace /etc/squid3/squid.conf:

```
http_port 80
http_port 443
auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid3/passwords
auth_param basic realm proxy
acl authenticated proxy_auth REQUIRED
http_access allow authenticated
```

  - Create a user

```
sudo htpasswd –c /etc/squid3/passwords username
```

Preparing Your Client

Client Setup

- On the hostile network you will not have Internet; get your laptop set up beforehand
- Ideally Linux/Kali, but Windows will work fine
  - Make sure your network driver supports MAC changing; most Windows drivers do not
  - Many USB network cards have great support for Windows (see Alfa Networks, Realtek, Atheros)
  - Can always run Linux/Kali in HyperV on Windows 8/10s
- Preinstall tools:
  - MobaXTerm (or any SSH client that supports tunneling; Linux has this built in)
  - Iodine (or any other IP-over-DNS tool; iodine is well-supported)
  - Wireshark on Windows; aircrack-ng on Linux
  - nmap
  - Fiddler2 on Windows; any HTTP debugging proxy on Linux. Charles if you're willing to shell out money.

Exploiting

Look Around

- Use ipconfig /all (ifconfig on Linux) to see your current IP

```
IPv4 Address. . . . . . . . . . . : 192.168.1.130
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 192.168.1.1
DNS Servers . . . . . . . . . . . : 192.168.1.1
```

- Use nmap to see what's there, and also check out the gateway

  nmap 192.168.1.0/24

  nmap 192.168.1.1 -A

- Looking for proxies (TCP/3128 is promising) and other unknown ports, also DNS (UDP/53)

Poke Around

- Try connecting to possible proxy ports (via browser config)
- Try connecting to your server (via HTTP or SSH) over port numbers open on the gateway
  - Yes, this shouldn't work.
  - Due to oddly configured transparent proxies, it sometimes does anyway.
- Try DNS lookups. If they succeed, look up your iodine domain.

Get Out

- If you have a route to a working proxy (gateway's or yours), you're done; configure browser.
- If you can SSH to your server, open a tunnel
  - Tools->MobaSSHTunnel on MobaXTerm
  - `ssh -L 8888:localhost:remoteport username@server.com` on Linux
  - Now you have a working local proxy; configure your browser
- If you can look up your iodine DNS, open a tunnel
  - `iodine -f -P password subdomain`
- Fix routing to point through the new tunnel
  - Route to your server's public IP goes through the existing gateway
  - New default gateway goes through the tunnel (172.16.0.0)

If All Else Fails

- Chilispot and its clones just configure iptables with MAC filters
- Use airodump-ng to watch traffic on the network
  - MACs with no traffic probably aren't authenticated
  - Squatting on a MAC currently in use will be a poor connection
  - Find a MAC with significant traffic that has stopped communicating
- Use macchanger to squat on the authorized MAC, then release/renew DHCP
- On Windows, can use Wireshark with filters instead of airodump-ng
  - Use Device Manager->Network Adapter->Advanced->Physical Address to change MAC
  - If not available, your WiFi driver does not support MAC changing; get a USB WiFi card down in the vendor room, something with an Atheros or Realtek chipset. You want one of these anyway.

Demonstrations

Scanning

DNS Tunneling

MAC Spoofing

#perimetergrid

Updated Slides at
http://perimetergrid.com/DefCon24.pptx